



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
[www.uspto.gov](http://www.uspto.gov)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/839,526	04/20/2001	Dietrich Charisius	BOR-006	2147
64107	7590	09/18/2009		
KOKKA & BACKUS, PC 200 PAGE MILL ROAD SUITE 103 PALO ALTO, CA 94306			EXAMINER MITCHELL, JASON D	
			ART UNIT	PAPER NUMBER
			2193	
			MAIL DATE	DELIVERY MODE
			09/18/2009 PAPER	

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

### Office Action Summary

**Application No.**

09/839,526

**Applicant(s)**

CHARISIUS ET AL.

**Examiner**

JASON MITCHELL

**Art Unit**

2193

**-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --**  
**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☒ Responsive to communication(s) filed on 10 July 2009.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) ☒ Claim(s) 1-4, 7-28, 31-65, 68-89 and 92-136 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-4, 7-28, 31-65, 68-89 and 92-136 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO/SB/08)  
Paper No(s)/Mail Date \_\_\_\_\_
- 4) ☐ Interview Summary (PTO-413)  
Paper No(s)/Mail Date \_\_\_\_\_
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: \_\_\_\_\_

### **DETAILED ACTION**

This action is in response to a request for continued examination filed on 7/10/09.

Claims 1-4, 7-28, 31-65, 68-89 and 92-136 are pending in this application.

### ***Response to Arguments***

**Applicant's arguments filed 7/10/09 have been fully considered but they are not persuasive.**

#### Independent Claims

In the last full par. on pg. 36, the applicants state:

As amended, claim 1 recites ... "generating a language-neutral representation of the source code modification; modifying and displaying via the computer a graphical representation of at least a portion of the language-neutral representation to reflect the source code modification and a textual representation of at least a portion of the language-neutral representation to reflect the source code modification, wherein the textual representation and the graphical representation are displayed simultaneously." Applicant respectfully submits that Walton, Chainini, and Hedin do not describe, disclose, teach, or suggest the limitation recited above.

The examiner respectfully disagrees. First, as indicated in the previous rejection, simultaneous display of textual and graphical representations would have been obvious to those of ordinary skill in the art. Further, the newly added "generating a language-neutral representation of the source code modification" is very broad in that it recites no details of this language neutral representation. On pg. 10, lines 10-21 the applicants describe the language neutral representation as a directory structure (e.g. "the language-neutral representation ... comprises SCI model 302 ... a directory for a software project ... SCI package 304 ... a subdirectory"). In view of this disclosure those of ordinary skill in the art would have understood the limitation to refer to any structure

other than simple source code in a particular programming language used to represent the program. Accordingly Walton's disclosure of a database for representing the objects and their relationships (*column 8, lines 44-65, "The resulting objects are then stored as objects in an object-oriented database system and connected to other objects or user code 120 in accordance with techniques commonly used in object-oriented systems."*) meets the broadly claimed limitation.

Additionally it is noted that use of a language neutral representation of a program is relatively common in the art, particularly where that program is intended to be represented in more than one format (see e.g. US 6,873,146 to Iyengar).

In the par. bridging pp. 36-37, the applicants state:

Moreover, it is respectfully submitted that there is no motivation to combine Walton with Chainini. Chainini teaches "a graphical programming application ... to be run under a graphical user interface." See abstract. The application can "activate a mode in which both code editors are visible at one time on the screen, so that changes made to the VBA statements are immediately reflected in the graphical program and vice versa." See abstract. Applicant respectfully submits that there is no motivation to combine Chainini with Walton, which teaches a "system for providing a simple, easy to learn and flexible means of creating user interfaces to products under development without the need of a programming language." See abstract. There is no motivation to combine a system that does not require the knowledge of programming with an implementation that is configured to simultaneously show a code editor and a graphics editor in one screen.

The examiner respectfully disagrees. Those of ordinary skill in the art would have been motivated to make the combination as a known alternative means of displaying the program in Walton's development environment (col. 7, lines 44-47 "A visual software engineering system") which would provide additional flexibility and ease the knowledge burden of developers while providing learning opportunities (Chainini col. 3, lines 50-54

"enables a user to design and modify a graphical program, and in the process, assists the user in learning to program a computer").

***Claim Rejections - 35 USC § 101***

35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

**Claims 62-65, 68-89 and 92-122 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.**

**Claims 62-65, 68-89 and 92-122** are not limited to statutory embodiments. In view of Applicant's disclosure, specification page 11, line7-12, the claimed medium is not limited to statutory embodiments, instead being defined as including both statutory embodiments (e.g., "hard disks") and non-statutory embodiments (e.g., "carrier wave"). As such, the claim is not limited to statutory subject matter and is therefore non-statutory.

***Claim Rejections - 35 USC § 103***

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

Claims 1-4, 7-15, 17-28, 31-32, 34-65, 68-76, 78-89, 92-93 and 95-136 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Walton** et al. (USPN 5,883,639) in view of **Chainini** et al. (USPN 5,760,788) in view of **Hedin** ("Supporting Programming Conventions").

### **Claim 1**

**Walton** disclosed a method in a data processing system including a computer, comprising the steps of:

providing the computer with a software development tool having a user interface that is operable by a user to automatically reflect a modification in the source code to avoid completely regenerating the source code (*figure 1 and 17; the software tool/system and its execution environment*), wherein the software development tool includes computer instructions for performing the following (*column 8, lines 44-65, "The interface designer next creates and/or modifies the graphical objects in the drawing and behavior editor 110 as desired using the available functions of the graphics editor of the designer's computer system. The resulting objects are then stored as objects in an object-oriented database system and connected to other objects or user code 120 in accordance with techniques commonly used in object-oriented systems."*) steps:

receiving an identification of a data structure with an attribute field in a database of data structures useable to form an object-oriented element from the data structure (*figure 1, elements 100 and 110, selected components are data structures with attributes in a database, 100; column 8, lines 44-65, object-oriented code development*);

determining via the computer whether the data structure is associated with source code (*figure 1, elements 120 and 130, source code developed base on object components*);

when a determination is made that the data structure is associated with the source code, another determination is made via the computer as to whether the attribute field of the data structure is associated with an attribute in the source code (*figure 1, elements 120 and 130, source code developed base on object components; column 8, lines 58-62*);

when an alternative determination is made that the attribute field is not associated with an attribute in the source code, generating via the computer a new attribute in the source code from the attribute field (*figure 1, elements 120 and 130, source code developed base on object components*); and

receiving user input to modify the source code (*figure 1, elements 120 and 130, source code developed base on object components; column 8, lines 44-65, "The interface designer next creates and/or modifies the graphical objects in the drawing and behavior editor 110 as desired using the available functions of the graphics editor of the designer's computer system. The resulting objects are then stored as objects in an object-oriented database system and connected to other objects or user code 120 in accordance with techniques commonly used in object-oriented systems."*);

generating a language-neutral representation of the source code modification (*column 8, lines 44-65, "The resulting objects are then stored as objects in an object-oriented database system and connected to other objects or user code 120 in*

*accordance with techniques commonly used in object-oriented systems.”; note that the applicants specification describes the language-neutral representation as a ‘directory’ comprising ‘subdirectories’, accordingly Walton’s database reasonably meets the broadly claimed ‘language-neutral representation’ );*

modifying and displaying via the computer a graphical representation of at least a portion of the language-neutral representation to reflect the source code modification (*column 9, lines 12-17 “create, delete and manipulate objects ... The resulting graphics images are displayed on a display 330”; column 8, lines 51-62; thus the user manipulating the graphical objects alters the code which is “graphically represented” through the graphical objects*) and a textual representation of at least a portion of the language-neutral representation to reflect the source code modification (Fig. 1, 120 & 130),; and

**Walton** does not explicitly disclose the textual representation and the graphical representation are displayed simultaneously (although it is noted that Fig. 1 could be understood to represent a single display screen of the VSE system and would then show source, 120 & 130 and graphical code, 110, displayed simultaneously).

**Chainini** teaches displaying a textual representation and corresponding graphical representations simultaneously (*col. 4, lines 26-34 “simultaneously view both an editor for the graphical program and an editor for the text-based programming language. As a result, changes in the graphical program are immediately reflected in the corresponding*



*commands or the text-based programming languages and changes in the commands of the text-based programming language are immediately reflected in the corresponding graphical program").*

It would have been obvious to one of ordinary skill in the art at the time the invention was made to display **Walton's** graphical (*Fig. 1, 110*) and textual code (*Fig. 1, 120, 130*) simultaneously as taught by **Chainini** (*col. 4, lines 26-34 "simultaneously view both"*). Those of ordinary skill in the art would have been motivated to do so as a known alternative means of implementing **Walton's** development environment (*col. 7, lines 44-47 "A visual software engineering system"*) which would provide additional flexibility and ease the knowledge burden of developers while providing learning opportunities (**Chainini** *col. 3, lines 50-54 "enables a user to design and modify a graphical program, and in the process, assists the user in learning to program a computer"*).

**Walton** and **Chainini** do not teach using a QA module to evaluate the modified source code, wherein an error message is generated and displayed if the modified source code does not conform to a predefined or user-defined style.

**Hedin** teaches using a QA module to evaluate the modified source code, wherein an error message is generated and displayed if the modified source code does not conform to a predefined or user-defined style (*pg. 3, section 3.1, 1<sup>st</sup> par. "Special error-attributes are used to give feedback to the programmer if restrictions are violated"*).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to include a programming convention tool as taught by **Hedin** (pg. 2, section 2, last par. “tools which make both the definition and application of conventions simple and natural”) in the development environment of the **Walton-Chainini** combination (e.g. **Walton** col. 7, lines 44-47 “A visual software engineering system”). Those of ordinary skill in the art would have been motivated to do so “to promote “good style”, making code easier to understand and maintain” (**Hedin** pg. 2, section 2, par. 5<sup>th</sup> par.).

## Claim 2

**Walton** disclosed the method of claim 1, further comprising: the steps of: when it is determined that the data structure is not associated with source code, retrieving a portion of the data structure; and generating the source code from the portion of the data structure (column 8, lines 54-62; figure 1, elements 120 and 130; producing code from library/database of components).

## Claim 3

**Walton** disclosed the method of claim 1, further comprising the steps of:

when it is determined that the data structure is associated with source code, determining whether a second attribute in the source code is associated with a second attribute field in the data structure (figure 1, multiple components; figure 17); and

when it is determined that a second attribute in the source code is not associated with a second attribute field in the data structure, removing the second attribute from the source code (*column 9, lines 13-15, "delete" components and thus code*).

#### **Claim 4**

**Walton** disclosed the method of claim 3, wherein the step of removing the second attribute from the source code comprises the step of removing a method associated with the second attribute from the source code (*column 9, lines 13-15, "delete" components and thus code, including associated methods*).

#### **Claim 7**

**Walton** disclosed the method of claim 5, further comprising the step of modifying the graphical representation of the source code to reflect the removal of the second attribute (*column 9, lines 13-15, "delete" components and thus code*).

#### **Claim 8**

**Walton** disclosed the method of claim 1, wherein the step of determining whether the data structure is associated with the source code comprises the step of searching a comment in the source code for the identification of the data structure (*figure 2, note comments*).

### **Claim 9**

**Walton** disclosed the method of claim 1, wherein the step of determining whether the data structure is associated with the source code comprises the step of comparing a name for the source code with the identification of the data structure (*figure 2, note comments*).

### **Claim 10**

**Walton** disclosed the method of claim 1, further comprising the steps of: retrieving access information for the database; and retrieving a portion of the data structure from the database using the access information (*figure 1, element 100 and 130*).

### **Claim 11**

**Walton** disclosed the method of claim 10, wherein the step of retrieving the access information comprises the step of retrieving the identification of the data structure and the access information from a configuration file (*figure 1, element 100 and 130*).

### **Claim 12**

**Walton** disclosed the method of claim 10, wherein the step of retrieving the access information comprises the step of retrieving the identification of the data structure and

the access information from a comment of the source code (*figure 1, "include" statement also performs a commenting function*).

### **Claim 13**

**Walton** disclosed the method of claim 10, wherein the portion of the data structure comprises the attribute field of the data structure (*figure 1, elements 100 and 110, selecting and manipulating components*).

### **Claim 14**

**Walton** disclosed the method of claim 1, wherein the source code comprises a class (*column 8, lines 54-56*).

### **Claim 15**

**Walton** disclosed the method of claim 1, wherein the source code comprises a distributed computing component (*column 8, lines 54-56; distributed as far as from a database*).

### **Claim 17**

**Walton** disclosed the method of claim 1, wherein the step of generating the new attribute in the source code comprises the step of generating a method in the source

code to access the attribute field of the data structure (*column 8, lines 54-56 "connected to other objects"*).

**Claims 18-28, 31-32, 34-65, 68-76, 78-89, 92-93 and 95-136**

The limitations of claims 18-28, 31-32, 34-65, 68-76, 78-89, 92-93 and 95-136 correspond to the limitations found in method claims 1-4 and 7-17 and are rejected in the same manner.

***Claim Rejections - 35 USC § 103***

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

Claims 16, 33, 77 and 94 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Walton** et al. (USPN 5,883,639) in view of **Chainini** et al. (USPN 5,760,788) in view of **Hedin** ("Supporting Programming Conventions") in view of Official Notice.

**Claims 16, 33, 77 and 94**

**Walton** discloses the method of claim 15, but does not explicitly disclose the distributed computing component is an Enterprise JavaBean.TM.

Official Notice is taken that it was known at the time of invention to make use of JavaBean components.

It would have been obvious to one of ordinary skill in the art at the time of invention to implement the components of **Walton** with including Enterprise JavaBean components. This implementation would have been obvious because one of ordinary skill in the art would be motivated to make use of all components on the market in-order to reach the largest available clientele.

### ***Conclusion***

Any inquiry concerning this communication or earlier communications from the examiner should be directed to JASON MITCHELL whose telephone number is (571)272-3728. The examiner can normally be reached on Monday-Thursday and alternate Fridays 7:30-5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Bullock Lewis can be reached on (571) 272-3759. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Jason Mitchell/  
Primary Examiner, Art Unit 2193